

KSH Államigazgatási Számítógépes Szolgálat

A komplex országos morbiditási vizsgálat /KOMOV/ számítás-
technikai vetülete

Molnár László, dr. Széphalmi Géza,

Bácsfalviné dr. Tóth Erika

A KOMOV /Komplex Országos Morbiditási Vizsgálat/ előkészítése már évekkel ezelőtt megkezdődött. Elsődleges célja: adatok szolgáltatása az egészségügy következő ötéves tervének megalapozásához. Négy különböző mintában összesen mintegy 50 ezer embert fog át. A munkálatokat az EüM koordinálja, az egyes minták házigazdái az OTKI és a KSH, illetve a SZOTE, POTE és a DOTE szervezési intézetei. A feladat számítástechnikai kivitelezésére az ÁSZSZ vállalkozott. 1977 közepétől kezdve veszünk részt a munkában, de még mindig nem fejeződtek be maradéktalanul az előkészületek. A felmérés a már véglegesített bizonylatok alapján ez év elején megkezdődött, az adatfelvétel időtartama egy év.

Számítástechnikai szempontból a következő információk határozzák meg a feladat megoldását:

1./ Az adatfelvétel nagyon kiterjedt, mintegy 10-12 kérdő-

ivtipust használnak. Szerepelnek közöttük demográfiai adatok /foglalkozás, iskolai végzettség, szociális környezet, stb./, szűrővizsgálati /kontrol/ adatok, követéses vizsgálati adatok /rejtett morbiditás feltárására/, járó- és fekvőbetegellátási adatok, szociológiai adatok, retrospektív adatok, stb.

- 2./ A kérdőíveken a felmérés során várhatóan mintegy 150 millió karakter információ gyűlik össze.
- 3./ A felmérés időszakában nincs feldolgozási igény, utána viszont rendkívül széles skálájú, több szintű, az első fázisban többnyire statisztikai jellegű igénylistát kell kielégítenünk, ugyanakkor fel kell készülnünk ad hoc kérdések megválaszolására is.
- 4./ Az egy éves adatfelvétel befejeztével még egy rövid idő a javításokkal telik el, de ezután a meglévő adatok már nem módosulnak és új adat sem érkezik.
- 5./ Az egyes bizonylat típusok között tartalmi kapcsolatok vannak.
- 6./ Hosszabb távon igény van az adatok interaktív - és akár egyidőben több felhasználó által - történő lekérdezésére is.
- 7./ Kapacitáshiány miatt az adatrögzítést alvállalkozóval

végeztetjük. Itt történik meg az adatellenőrzés és a javítás is, de csak "kérdőíven belüli" ellenőrzésekre kerül sor. Ezzel a módszerrel a javítások befejezése után komplex és az említett szempontból "hiteles" adathalmazt kapunk.

8./ Érdemes külön kiemelni a követéses /follow-up/ vizsgálatot, ezzel kapcsolatban figyelmet érdemel az a kérdés, hogyan lehet célszerű módon tárolni a követéses adatokat fix méretű rekordok alkalmazásával.

A fentiek alapján adatbázis kialakítása ellen szólnak a következő érvek:

- az adatállomány a mi szempontunkból statikus jellegű, tehát nem használjuk ki az adatbázisok által nyújtott egyik legfontosabb előnyt;
- bonyolultabb struktúra tovább növeli a tároláshoz szükséges terület méretét és növelheti a feldolgozás időszükségletét;
- általános, statisztikai jellegű feldolgozások futtatása közvetlenül az adatbázisból vett adatok alapján nagyon időigényes lehet, különösen akkor, ha sok különféle igényt kell kielégíteni /az adatbázis strukturája önmagában nem biztosíthatja a kellő mértékű rugalmasságot/.

Az adatbázis kialakítása mellett szóló érvek:

- az adatbázis strukturája jó lehetőséget biztosít a kérdőívek közötti kapcsolatok ellenőrzésére, tehát a teljes adatállomány konzisztenciájának biztosítására;
- lehetővé teszi az interaktív, "konkurrens" lekérdezéseket;
- a rendkívül változatos, de mindig azonos adathalmazra épülő információ-igény miatt célszerű az adatokat és a feldolgozó programokat különválasztani.

Ugyanakkor az adathalmaz statikus jellege is meghatározó az adatbázis kialakításánál:

- bonyolultabb strukturát is ki lehet alakítani /ami nem az adatok szétforgácsolását jelenti, hanem azok többszemponthu csoportosítását/;
- lényegében nem fordulhatnak elő az ún. tárolási, módosítási, törlési "anomáliák";
- az adatbázis fizikai szintjén is lehet redundáns adatokat használni, mert nem fenyeget az inkonzisztencia veszélye;
- az adatbázis töltésekor, a feldolgozások meggyor-

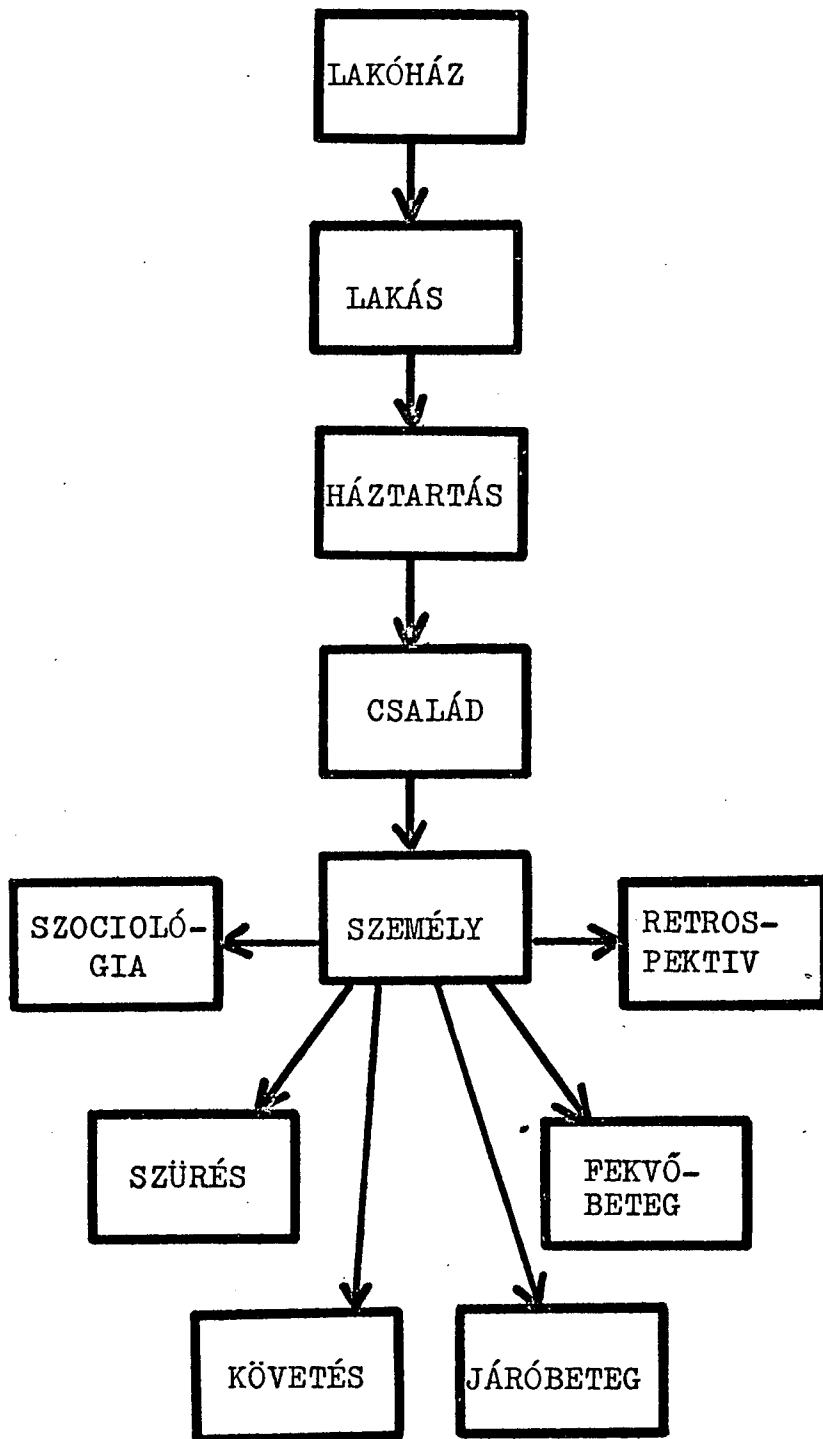
sításához célszerű bizonyos "generált" adatokat előre kiszámítani és azokat tárolni, mert ezek is csak elvétele változhatnak.

Mindezek alapján a hagyományos feldolgozással szemben adatbáziskezelő rendszer alkalmazása mellett döntöttünk. Erre a célra a HWB IDS-I rendszere kínálkozott, mint az ÁSZSZ eszközbázisán jelenleg egyedüli lehetőség.

Az IDS-I a COSASYL típusu hálózatos adatbáziskezelő rendszerek előfutára, lényegében azoknak egy egyszerűsített változata.

A kérdőívek közötti kapcsolatot a következő struktúra alapján lehet ellenőrizni /1. ábra/. /Első közelítésben e szerint határoztuk meg az adatbázis logikai felépítését, amit aztán a normalizálási eljárás segítségével fogunk elemezni, majd pedig az output-igények és egyéb szempontok alapján módosítunk, kihasználva a statikus jelleg nyújtotta lehetőségeket is./

Minden feldolgozás menete a következő: bizonyos adatmezők tartalma alapján "csatolt" adatokat keresünk vissza, ezekkel valamilyen műveletet végzünk, majd az eredményt "láthatóvá" tesszük. Az egyik feladat tehát az, hogy lehetővé tegyük az adatbázisban szereplő rekordok minél sokrétűbb osztályozását. Az az elképzelésünk, hogy olyan prog-



1. ábra

ramrendszert alakítunk ki, amely ehhez az adatbázis strukturájának kihasználása mellett szükség esetén a nem rendezett mintán történő szekvenciális feldolgozási módban rejlő lehetőségeket is biztosítja. Ettől a kompromisszumtól a hatékonyság jelentős növekedését várjuk.

Az elmondott elv megvalósítására a következő lehetőségek állnak rendelkezésünkre:

1./ Az adatbázis strukturája által nyújtott lehetőség:

bizonyos mezők tartalma alapján a rekordokat egymáshoz láncolva tároljuk. Mivel ezt az adatbázis töltésekor kell végrehajtani, ez a csoportosítás "végleges". Ez a megoldás azért sem rugalmas, mert az IDS-ben a lánc szerinti olvasás nem paraméterezhető. Korlátot jelent az egyidejűleg figyelembe vehető mezőkombinációk száma is. Mindebből az következik, hogy ezt a lehetőséget csak az előre ismert fontosabb - és lehetőleg csak egy adatmező tartalma szerinti - felosztásokra célszerű alkalmazni.

2./ Az egyes rekordok tárolásakor a már említett "generált" adatok között képezhetünk és a rekordba illeszthetünk mezők szerinti osztályozásokat. Ezt célszerű átkódolt formában megtenni, mert ez a megoldás szekvenciális feldolgozással folytatódik és lényegében

egy táblázat kitöltését jelenti. Akkor gyors a feldolgozás, ha ez a táblázat a memóriában elfér. Ennek érdekében a memóriakapacitást többszöri olvasással is meg lehet növelni szükség esetén. /Kb. ötszöri olvasással még várhatóan nem lesz a futási időtöbblet túl nagy./ Ez a megoldás, bár a szükséges tárolóigényt növeli, az előzőnél rugalmasabb megoldást biztosít egyrészt azért, mert több szempont kombinációja szerinti osztályozást is lehetővé tesz, másrészt azért, mert az osztályozási szempontok kiválasztása paraméterezhető. A rugalmasság tovább növelhető, ha egymásba skatulyázott felosztásoknál a lehető legfinomabb felosztás szerinti osztályozást választjuk. Mivel azonban ezt a "generálást" is az adatbázis töltésekor célszerű elvégezni, az egyes osztályozások itt is véglegesek. Mindezek alapján ezt a módszert ott érdemes alkalmazni, ahol a feldolgozás több, előre ismert szempont szerinti együttes osztályozást kíván.

- 3./ A legrugalmasabb /paraméterezhető/, de viszonylag hosszabb feldolgozási időt igénylő megoldás az, amikor a rekordok osztályozását - legalábbis részben - magára a feldolgozó programra bizzuk. A módszer ilyenkor is szekvenciális olvasás és osztályozás, ill. átkódolás utáni táblázat-kitöltés. Ez a megoldás nyilván-

valóan az előre nem ismert, ad hoc osztályozások megállapítására alkalmas elsősorban.

4./ Az előzőektől eltérő "vegyes" lehetőséget biztosít az MDQS, az IDS adatbázisok egyik interaktív lekérdező rendszere. Segítségével kombinálni lehet az IDS struktúra által nyújtott lehetőségek kihasználását a másik két megoldással. Magasszintű nyelven röviden le lehet írni bonyolult szempontok alapján történő rekordkiválasztásokat, "virtuális" rekordok felépítését, a kiválasztott rekordok adataival végzendő műveleteket, és az eredmények kitáblázását is.

Az elmondottakat összefoglalva az a célunk, hogy a feladat megoldására adatbázist hozzunk létre, de - kihasználva az adatok statikus jellegét - úgy, hogy az adatbázis struktúrája, illetve az adatbáziskezelő és lekérdező rendszer által nyújtott lehetőségeken túl megteremtsük a gyors szekvenciális feldolgozások lehetőségét is és ezáltal megnöveljük a rendszer hatékonyságát.